Instituto Politécnico de Beja

Escola Superior de Tecnologia e Gestão

Mestrado em Engenharia de Segurança Informática

Implementação de Cifras Caóticas sobre Caos Modular

Orientador: Professor Doutor Rui Miguel Silva

Discente: Bruno Filipe Elias Dias, nº 14697

Beja, 24 de novembro de 2017

Instituto Politécnico de Beja

Escola Superior de Tecnologia e Gestão

Mestrado em Engenharia de Segurança Informática

Curso: Mestrado em Engenharia de Segurança Informática

Título: Implementação de Cifras Caóticas sobre Caos Modular

Dissertação de Mestrado apresentada na Escola Superior de Tecnologia e Gestão do Instituto Politécnico de Beja

Orientador: Professor Doutor Rui Miguel Silva

Discente: Bruno Filipe Elias Dias, nº 14697

Beja, 24 de novembro de 2017

RESUMO

Nesta dissertação, foi realizado um estudo aprofundado sobre Criptografia, Sistemas Caóticos, Cifras Caóticas, Caos Modular e um estudo final exaustivo do funcionamento da Cifra Caótica eLoBa.

O objetivo desta dissertação foi implementar e melhorar a estrutura da Cifra Caótica eLoBa em outros sistemas caóticos além do Sistema Caótico de Lorenz, de modo a que, depois de implementados, foram recolhidas 1 milhão de chaves de cada Sistema Caótico para cada uma das duas sementes com apenas um caractere diferente.

No final, são apresentados os resultados em tabelas e gráficos dos dados recolhidos da Bateria de Testes Estatísticos a Números Pseudo-aleatórios, NIST 800-22, com as amostras dos Sistemas Caóticos.

De acordo com os dados recolhidos ficou confirmado que é possível a aplicação dos Sistemas Caóticos em Criptografia.

Palavras-chave: Criptografia, Chaves, Cifras, Sistemas Caóticos, Nist 800-22

ABSTRACT

In this work an in - depth study was carried out on Cryptography, Chaotic Systems, Chaotic Figures, Modular Chaos and a final exhaustive study of the functioning of the Chaotic Cipher eLoba.

The objective of this work was to implement and improve the structure of the Chaotic Cipher eLoBa in other chaotic systems besides the Chaotic System of Lorenz, so that once implemented, 1 million keys were collected from each Chaotic System for each of the two seeds with only a different character.

At the end, the results were presented in tables and graphs of the information collected from the Battery of Statistical Tests to Pseudo-Random Numbers, NIST 800-22, with the samples of the Chaotic systems.

According to the data collected it was confirmed that it is possible to apply the Chaotic Systems in Cryptography.

Keywords: Cryptography, Keys, Ciphers, Chaotic Systems, Nist800-22

AGRADECIMENTOS

Os meus agradecimentos a todos os que possibilitaram a realização desta dissertação nomeadamente:

Ao meu orientador, Professor Doutor Rui Miguel Silva, por me auxiliar sempre que precisei;

Aos meus pais, Ilda e Francisco, e irmã, Isa, que me apoiaram incondicionalmente;

À minha namorada e melhor amiga, Savannah Salgueiro, por estar sempre ao meu lado e por todo o apoio e força de vontade que me tem emprestado para conseguir chegar até aqui,

Aos meus amigos, Diogo Bentes, José Santos e Tobias Cintrão pela ajuda técnica que existiu da sua parte;

Deixo os meus sinceros agradecimentos a todos os que não foram citados, mas que, de alguma forma, auxiliaram na realização deste trabalho.

ÍNDICE

LISTA DE ABREVIATURAS

AES (*Advanced Encryption Standard*)

Cifra Caótica eLoBa (*enhanced Lorenz Based*)

GMP (*GNU Multiple Precision Arithmetic Library*)

IP (*Internet Protocol*)

LFSR (*Linear-Feedback Shift Register*)

LFSR_CD (*LFSR_ChaoticDisturbance*)

LFSR_KM (*LFRS_KeyMixture*)

NIST (*National Institute of Standart and Tecnology*)

NIST 800-22 (*Statistical Test Suite for Random and Pseudo-Random Number Generators*)

XOR (*eXclusive OR*)

ÍNDICE DE FIGURAS E GRÁFICOS

ÍNDICE DE TABELAS

1. INTRODUÇÃO

Hoje em dia, cada vez mais, a tecnologia é a base de tudo na nossa vida quotidiana. Devido a esse facto e a uma evolução extremamente rápida da tecnologia é muito difícil conseguir com que a segurança informática acompanhe essa evolução. [1]

A segurança informática é necessária em tudo o que envolva privacidade e confidencialidade, quer seja pessoal, empresarial, médica entre outras. Sem proteger os dados informático corremos o risco de sairmos lesados de alguma maneira. [1]

Um dos grandes desenvolvimentos e evolução. a nível financeiro, foi a BitCoin, que, em menos de uma década tornou-se uma moeda digital comercial para realizar compras em qualquer sítio, existindo até cartões de multibanco para funcionarem com as mesmas. A sua evolução foi de tal modo grande e rápida que assusta bancos e bolsas, ao ponto de estes ficarem com receio do que poderá afetar no futuro. Mas, acima de tudo, se não fosse o sistema de encriptação, identificação e mineração das BitCoins tão bom e avançado como é, este já teria desaparecido e nunca teria evoluído a este ponto. [4, 5]

A segurança das comunicações, a qualquer nível, é uma área de constante evolução, com importância vital para todas as áreas com transição de informação sensível. As principais vertentes da melhoria desta segurança, são a criação de sistemas mais avançados e resilientes e, além disso, a procura de fragilidades nesses mesmos sistemas de modo a ficarem mais fortes. [2]

Existem muitos sistemas de encriptação e segurança da informação das comunicações, mas, a cada dia que passa, esses mesmos sistemas ficam mais perto de serem corrompidos, quebrados e/ou anulados. Não é uma motivação o querer desenvolver novos sistemas de encriptação, mas sim, uma necessidade mundial para cada um de nós. [3]

Por isso utilizar sistemas matemáticos complexos na encriptação de dados pode ser uma mais valia, e, quanto mais complexo e confuso, melhor. Devido às propriedades intrínsecas dos Sistemas Caóticos, estes poderão ser extremamente viáveis para a criação de sistemas criptográficos, pois também possuem um custo computacional muito baixo, sendo o mesmo a baixa utilização de recursos do computador. Logo existe a hipótese de construção de cifras baseadas em sistemas caóticos direcionada para a encriptação de comunicações ou de dados. [6]

1.1. Contextualização

Esta dissertação tem como principal objetivo o estudo das Cifras Caóticas, mais especificamente, a Cifra Caótica eLoBa (*enhanced Lorenz Based*) [7], para que posteriormente seja possível realizar uma reimplementação da estrutura da mesma para o sistema de Lorenz e outros Sistemas Caóticos que respeitem as mesmas condições necessárias.

Depois de realizada a escolha dos sistemas a implementar, foi feita uma seleção de que tamanho deveria ter o espaço de amostra de chaves recolhidas de cada sistema e quantas sementes (chaves secretas) seriam necessárias para realizar uma comparação de resultados. Definiram-se duas sementes com apenas 1 caractere diferente e o espaço de amostra de 1 milhão de chaves para cada semente.

No fim da recolha de dados, feita foi realizado um conjunto de testes estatísticos com a bateria NIST 800-22 (*Statistical Test Suite for Random and Pseudo-Random Number Generators*), para testar a viabilidade da implementação de Cifras Criptográficas com base em Sistemas Caóticos.

1.2. Estrutura da Dissertação

Esta dissertação é constituída por cinco capítulos, sendo este capítulo o primeiro destinado à **Introdução**.

No capítulo 2 são abordadas a **Criptografia e Sistemas Caóticos,** no qual são definidos os conceitos base da dissertação e apresentada a cifra caótica de estudo para esta dissertação, a Cifra Caótica eLoBa.

No capítulo 3, denominado **Caos Modular**, são apresentadas a arquitetura utilizada, com as justificações de desenho e a sua funcionalidade total e a de cada um dos seus sub-sistemas.

No capítulo 4, intitulado **Implementação de Cifras Caóticas**, serão apresentados critérios de seleção de Sistemas Caóticos, os sistemas escolhidos e os rejeitados, e a forma de recolha da amostra de dados.

No capítulo 5, designado **Avaliação e Comparação de Resultados**, é apresentada a informação de baterias de testes, os procedimentos para realização dos testes e, por fim, os resultados finais da análise aos Sistemas Caóticos selecionados.

Finalmente, no capítulo 8 apresentam-se a **Interpretação de Resultados e Conclusões finais**.

2. CRIPTOGRAFIA E SISTEMAS CAÓTICOS

2.1. Criptografia

A criptografia [8, 35] é o estudo dos princípios e técnicas pelos quais a informação pode ser transformada em uma série de caracteres aleatórios, sem qualquer sentido à primeira vista, por quem não tem qualquer conhecimento do que está apresentado no resultado final do que foi cifrado.

O principal objetivo da criptografia é esconder qualquer informação que seja transferida de um emissor para um recetor, de modo a que esta seja impercetível para qualquer pessoa, ou meio, que intercete a mensagem, ou seja, manter a confidencialidade, integridade, autenticação e não repúdio.

Existem dois tipos de criptografia [8, 35], a criptografia por chaves simétricas, em que existe apenas uma chave para o emissor e para o recetor, e a criptografia por chaves assimétricas, em que funciona com um sistema de chaves públicas e privadas.

Hoje em dia, a criptografia já não é apenas um estudo da cifragem e decifragem, mas sim, um ramo especializado da teoria de informação com contribuições também de outros campos da matemática e do conhecimento.

2.1.1. Algoritmo de Chave Simétrica

Os algoritmos de chave simétrica são um tipo de algoritmos que funcionam apenas com uma chave, tanto recetor e emissor na comunicação usam a mesma chave para cifrar e decifrar as mensagens.

É um sistema muito simples, em que o emissor e o recetor entram em acordo relativamente à chave secreta a utilizar e assumem que esta é apenas conhecida pelos dois, e de seguida inicia-se a troca de mensagens cifradas com o algoritmo de chave simétrica, como observado na figura 1. [38]

Figura 1: Chaves Simétricas [38]

2.1.2. Algoritmo de Chave Assimétrica

O algoritmo de chaves assimétricas é um conjunto de protocolos baseados em algoritmos que exigem a existência de duas chaves, em que uma é a chave privada/secreta, que serve para autenticar a informação enviada com uma assinatura digital e para decifrar a informação recebida de outrem, e a outra é a chave publica que faz exatamente o oposto da chave privada, cifrar a informação pretendida para a transmissão (figura 2) e verificar a autenticidade da informação recebida (figura 3). [34]

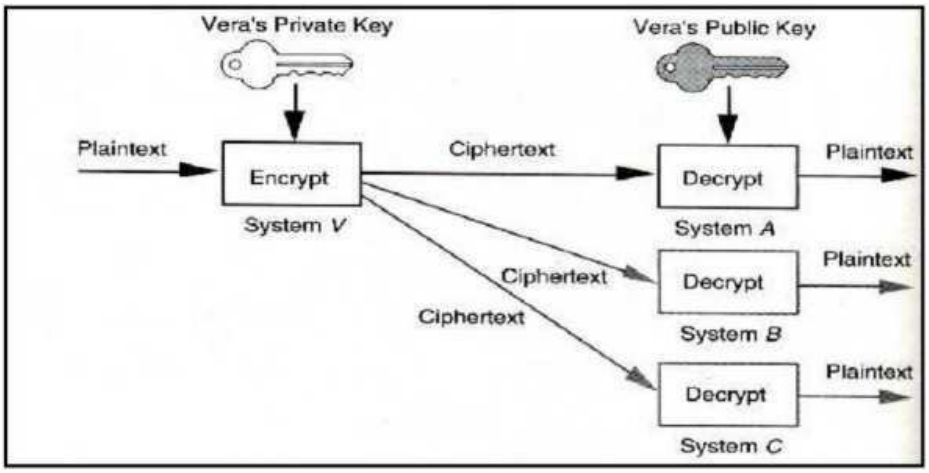

Figura 2: Chaves Assimétricas – Transmissão [34]

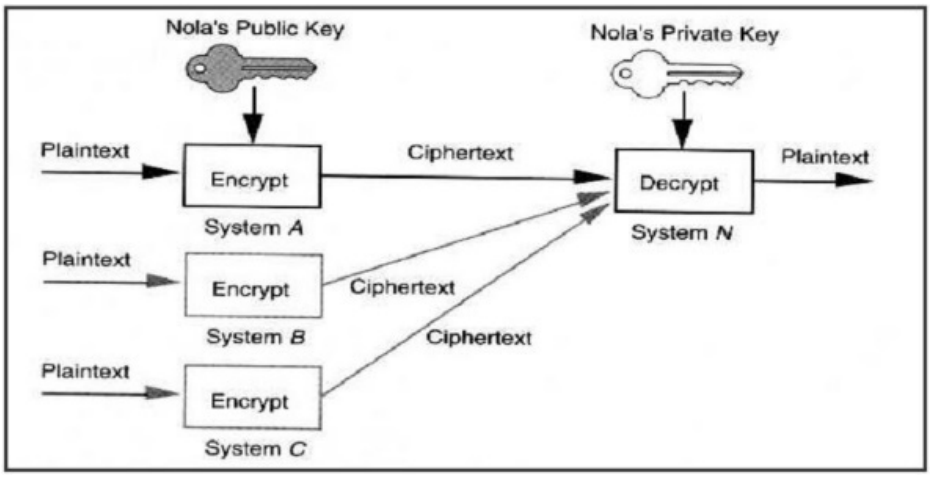

Figura 3: Chaves Assimétricas – Receção [34]

2.2. Sistemas Caóticos

Os sistemas caóticos são sistemas complexos e dinâmicos rigorosamente determinísticos, tendo uma propriedade fundamental, a instabilidade, que está dependente das condições iniciais que, em conjunto com a sua propriedade de recorrência, torna-os completamente imprevisíveis na prática a longo prazo. [6, 22]

Para melhor compreensão dos mesmos, pode-se usar um exemplo da natureza, onde estes sistemas são mais comuns. Uma nuvem no céu pode se formar e desenvolver com base em centenas de fatores como o calor, frio, evaporação da água, ventos, clima, condições do sol, entre outros, sendo, deste modo, muito difícil fazer previsões do tempo, acontecendo frequentemente as mesmas estarem erradas.

Um exemplo de Sistemas Caóticos é o comportamento de um fractal, como demonstrado na figura 4. [39]

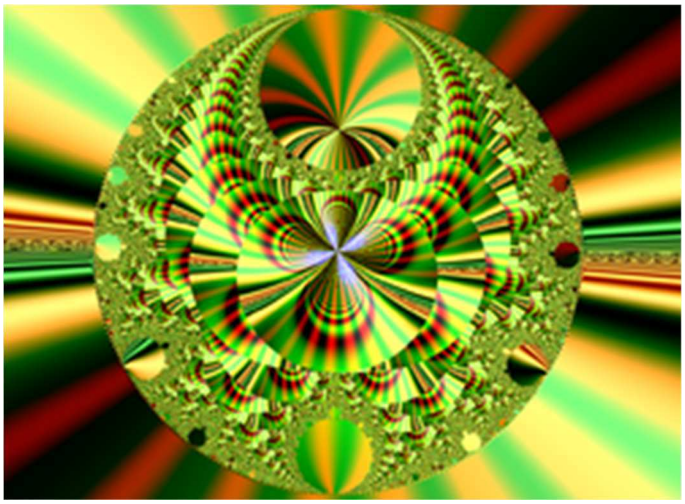

Figura 4: Fractal [39]

Os Sistemas Caóticos podem ser contínuos ou discretos, conforme os parâmetros abaixo explicados.

2.2.1. Contínuos

Um sistema dinâmico continuo é um sistema cujo estado evolui ao longo do tempo continuamente, ou seja, não existem pontos, está sempre em evolução, evoluindo durante a mudança de instantes $\{t_0, t_1, t_2, \ldots\}$. [24, 26]

2.2.2. Discretos

Um sistema dinâmico discreto é um sistema cujo estado só muda durante os instantes $\{t_0, t_1, t_2, \ldots\}$. Isto é, de um instante para o outro, nada acontece, não existe evolução. [27]

2.2.3. Contínuos Discreteados

Um sistema dinâmico discreteado é um sistema dinâmico continuo em que lhe foi aplicado o método de Euler. Após a aplicação deste método o sistema dinâmico continuo

funciona com um passo de integração a cada instante, iteração. Cada iteração é calculada com os valores da iteração anterior. Ou seja, o sistema dinâmico continuo com a aplicação do método de Euler serve para trabalhar com números inteiros apenas em sistemas dinâmicos contínuos como se fosse um sistema dinâmico discreto. [30]

2.3. Cifras Caóticas

Cifras Caóticas são sistemas de encriptação baseados em sistemas caóticos dinâmicos discreteados. Ou seja, é a utilização de sistemas dinâmicos contínuos onde foi aplicado o método de Euler, de modo a que se mantenham as propriedades dos sistemas dinâmicos contínuos, mas possuindo a fácil manipulação dos sistemas discretos.

2.3.1. Cifra Caótica eLoba – Enhanced Lorenz Based

A Cifra Caótica eLoBa foi criada com base em sistemas caóticos, especializada para redes sem fios, Wi-Fi. Esta foi melhorada de modo a aumentar o desempenho e segurança, nomeadamente contra ataques algébricos. [7]

A Cifra Caótica eLoBa devido ao seu baixo custo computacional consegue ser 40% mais rápida que o AES (*Advanced Encryption Standard*) no modo contador, além disso também diminui a sobrecarga de pacotes IP (*Internet Protocol*) do protocolo de transporte de redes sem fios. [29, 48, 49]

Esta cifra funciona do seguinte modo, entra na cifra uma chave secreta de 16 caracteres e uma mensagem, passa pela arquitetura de Caos Modular (explicado no capítulo 3) e à saída obtemos a mensagem encriptada por duas chaves de cifra sempre diferentes, cada uma com 128 bits.

Foi feito um estudo aprofundado da arquitetura desta Cifra Caótica para a realização desta dissertação.

3. CAOS MODULAR

3.1. Módulos da Arquitetura

Com o estudo da estrutura da Cifra Caótica eLoBa, ficou claro que, para o seu bom funcionamento são necessários 4 módulos, cada um com a sua função, mas em conjunto funcionam para garantir a segurança e a aleatoriedade da cifra. Assim, esta não entra em ciclos de vida curtos e mantém uma entropia entre o estado evolutivo do sistema e a chave de inicialização do sistema.

Os 4 módulos são:

- **Módulo do Sistema Caótico**, que contém o estado do sistema caótico aplicado com o Método de Euler e de parametrização fixa de modo a que o seu funcionamento permaneça com resultados aleatórios, caóticos;
- **Módulo de Perturbação Caótica**, que serve para criar alterações de órbita de forma a evitar ciclos de vida curtos ou pontos fixos, pelo facto de este funcionar discreteado;
- **Módulo de Mistura de Chaves**, que garante a segurança do estado interno dos sistemas caóticos, para as que chaves não fiquem expostas;
- **Módulo de Inicialização**, que controla todo o funcionamento do sistema criptográfico, de forma a que a desordem entre o sistema caótica e a chave de inicialização se mantenha do princípio ao fim. [7]

O Módulo de Perturbação Caótica tem duas propriedades que devem ser mantidas durante a encriptação. Estas são a periocidade das alterações que devem ser realizadas em todas as iterações do sistema e a forma de realizar as alterações em somas de módulo 2 entre o resultado de um LFSR (*Linear-Feedback Shift Register*) de ciclo completo e o valor das coordenadas, de modo a beneficiar das propriedades do XOR (*eXclusive OR*) e do LFSR. [33, 47]

Quanto à mistura de chaves, esta é feita do mesmo modo que o processo anterior, pelas mesmas razões. Além disso é feita uma conjugação dos 384 bits de cada iteração do Sistema Caótico, X, Y e Z, de forma a criar duas chaves de 128 bits e deixar mais 128 bits em segredo. A razão dos 128 bits em segredo é para segurança contra ataques que

descubram os 256 bits das chaves de cifra, mas, mesmo assim, como estas são alteradas a cada iteração, torna-se ainda mais difícil o ataque.

3.2. Estrutura da Arquitetura

A figura 5 apresenta a estrutura global da arquitetura da cifra. No interior da cifra encontra-se o gerador de números pseudoaleatórios constituído pelos módulos já referidos. [7]

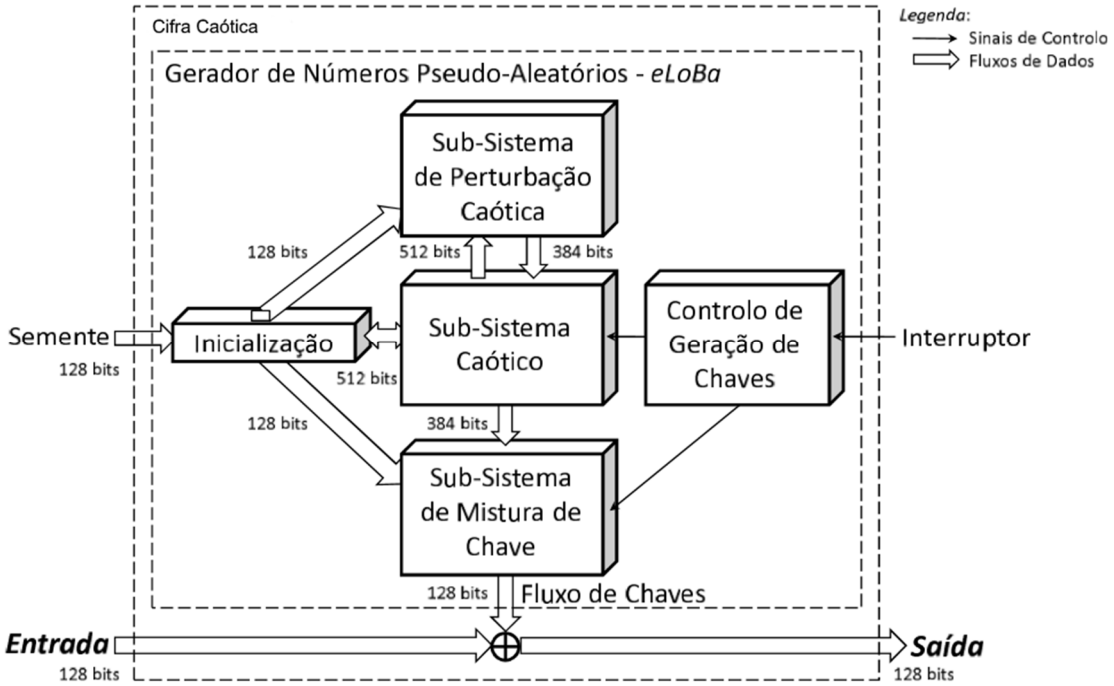

Figura 5: Arquitetura da Cifra Caótica [7]

3.3. Funcionamento da Arquitetura

3.3.1. Inicialização

O sistema é inicializado com a entrada da "semente" constituída por 128bits, 16 caracteres, e da mensagem, que poderá ter qualquer tamanho, mas apenas serão cifrados 128 bits de cada vez.

A "Inicialização" dos três sub-sistemas é feita sequencialmente recorrendo a um gerador pseudo-aleatório auxiliar formado por LFSR, a um conjunto de função de rotação de dados e à iteração do sistema caótico como elemento não linear.

O "Sub-Sistema Caótico" é iniciado com a atribuição do valor do estado inicial do LFSR à variável X_0, de seguida o LFSR é iterado mais três vezes, sempre com a iteração anterior, sendo o resultado de cada iteração, de 128 bits, os valores Y_0, Z_0 e Δt_0.

Na figura 6, é demonstrada a rotação que cada valor vai receber, respetivamente, rotação de 32 bits à direita, 32 bits à esquerda e 64 bits à direita, para que exista maior resistência a ataques algébricos. [7]

$$\begin{cases} LFSR_0 = Semente \\ X_0 = LFSR_0 \\ Y_0 = F1(LFSR_1) \\ Z_0 = F2(LFSR_2) \\ \Delta t_0 = F3(LFSR_3) \end{cases}$$

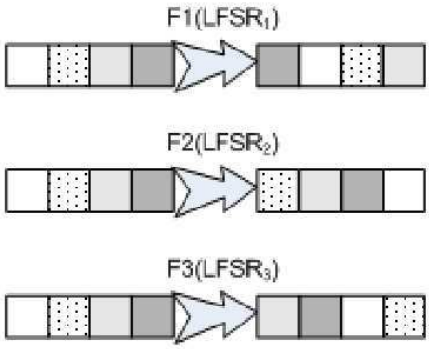

Figura 6: Sub-Sistema Caótico e Função de Rotação [7]

O segundo sub-sistema a iniciar é o "Sub-Sistema de Perturbação Caótica". Este é inicializado com uma iteração do sistema caótico com os valores X_0, Y_0 e Z_0 e o passo de integração Δt_0, calculando assim os novos valores X_1, Y_1 e Z_1. De seguida é aplicada a função de rotação nestes valores de 32 bits à direita como demonstrado na figura 7. Para finalizar será necessário criar o LFSR de perturbação caótica, com o nome LFSR_CD (*LFSR_ChaoticDisturbance*), como demonstrado também na figura 7. [7]

$$\begin{cases} LFSR\_CD_0[0:31] = X_1[32:63] \oplus Y_1[64:95] \oplus Z_1[96:127] \\ LFSR\_CD_0[32:63] = X_1[64:95] \oplus Y_1[96:127] \oplus Z_1[0:31] \\ LFSR\_CD_0[64:95] = X_1[96:127] \oplus Y_1[0:31] \oplus Z_1[32:63] \\ LFSR\_CD_0[96:127] = X_1[0:31] \oplus Y_1[32:63] \oplus Z_1[64:95] \end{cases}$$

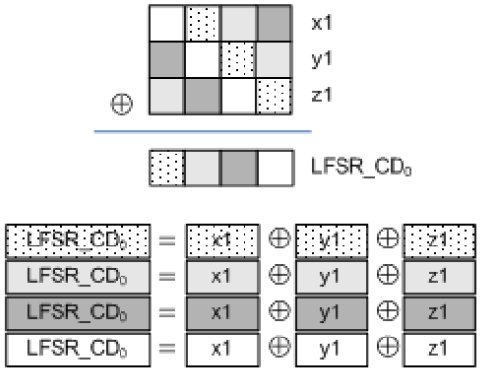

Figura 7: Sub-Sistema Perturbação Caótica e Função de Rotação [7]

Para finalizar a inicialização, temos o "Sub-Sistema de Mistura de Chave". Funciona igual ao anterior, mas agora este é inicializado com X_1, Y_1 e Z_1 e o passo de integração Δt_0, criando assim X_2, Y_2 e Z_2. A função é aplicada com 32 bits à esquerda, os valores X_2, Y_2 e Z_2 são enviados para a função LFSR para criar o LFSR_KM (*LFRS_KeyMixture*), como apresentado na figura 8. [7]

$$\begin{cases} LFSR\_KM_0[0:31] = X_2[96:127] \oplus Y_2[64:95] \oplus Z_2[32:63] \\ LFSR\_KM_0[32:63] = X_2[0:31] \oplus Y_2[96:127] \oplus Z_2[64:95] \\ LFSR\_KM_0[64:95] = X_2[32:63] \oplus Y_2[0:31] \oplus Z_2[96:127] \\ LFSR\_KM_0[96:127] = X_2[64:95] \oplus Y_2[32:63] \oplus Z_2[0:31] \end{cases}$$

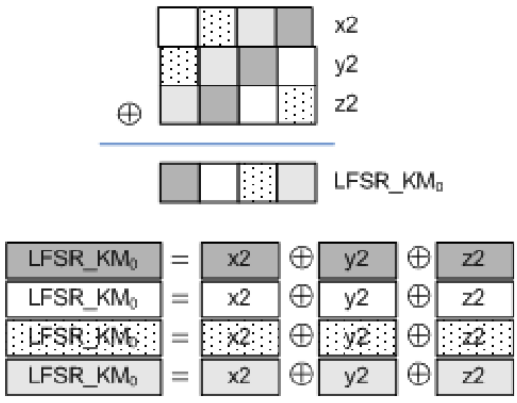

Figura 8: Sub-Sistema de Mistura de Chave e Função de Rotação [7]

3.3.2. Sub-Sistema Caótico

O "Sub-Sistema-Caótico" é implementado com as equações do sistema caótico e com a parametrização correta de modo a manter a funcionar o regime caótico do sistema. As equações são resolvidas matematicamente com o método de Euler, com uma aritmética modular de 128 bits com processamento exclusivo de números inteiros.

O uso do método de Euler faz com que os sistemas percam precisão, mas não tem qualquer problema quando o objetivo não é a precisão, mas sim a manutenção das propriedades caóticas para o funcionamento correto dos sistemas criptográficos.

Depois de inicializados todos os sub-sistemas, o "sub-sistema caótico" é o primeiro a entrar em ação. Este começa com a primeira iteração com os valores de X_2, Y_2 e Z_2 e o Δt_0 como passo de integração, ou seja, a cada integração i+1, o valor da iteração de i é somado também à iteração de i+1.

O valor de Δt_i funciona de um modo especial, entram 128 bits, mas apenas 8 bits são usados para realizar os cálculos, de modo a que seja possível poupar custo computacional, que seja mais rápido e eficiente a chegar ao resultado final. [7]

3.3.3. Sub-Sistema de Perturbação Caótica

Deste modo, depois de obtidos os valores de X_{i+1}, Y_{i+1} e Z_{i+1} e o Δt_i, o "Sub-Sistema de Perturbação Caótica" inicia a sua função. Este cria um novo $LFSR\_CD_{k+1}$ enviando o $LFSR\_CD_k$ para o modulo LFRS. Para depois realizar um "XOR" de Y_{i+1} com $LFSR\_CD_{k+1}$ resultando um Y_{i+1} modificado.

É realizado o mesmo procedimento com o $LFSR\_CD_{k+1}$ e assim gera um $LFSR\_CD_{k+2}$ que será calculado o "XOR" com Z_{i+1} para gerar um novo valor para Z_{i+1}.

Para finalizar será criado um Δt_{i+1} através de um "XOR" de X_{i+1} com Δt_i, e assim fica terminada a perturbação do sistema. [7]

3.3.4. Sub-Sistema de Mistura de Chaves

Para terminar o funcionamento da Arquitetura, o "Sub-Sistema de Mistura de Chaves" realiza a sua função. Este, por sua vez, recebe do sistema caótico as 3 coordenadas X, Y e Z, num valor total de 384 bits, para criar duas chaves de 128 bits e manter os restantes 128 bits secretos para segurança e proteção das chaves.

Antes da criação de cada chave, o $LFSR\_KM_k$ é sempre enviado para o módulo LFSR para a geração de um novo estado $LFSR\_KM_{k+1}$.

A criação da primeira chave é realizada com 96 bits de Y e 32 bits de X, a segunda chave é criada com 96 bits de Z e 32 bits de X. Depois da chave gerada, estas são calculadas num "XOR" com o valor do LFSR gerado antes de cada chave. Deste modo, 64 bits de X, 32 bits de Y e 32bits de Z mantêm-se secretos. [7]

A figura 9 ilustra o processo da criação das chaves.

$$\begin{cases} Key1 = X[24:31]||Y[8:31]||X[56:63]||Y[40:63]|| \\ \quad\quad X[88:95]||Y[72:95]||X[120:127]||Y[104:127] \oplus LFSR\_KM_i \\ Key2 = X[16:23]||Z[8:31]||X[48:55]||Z[40:63]|| \\ \quad\quad X[80:87]||Z[72:95]||X[112:119]||Z[104:127] \oplus LFSR\_KM_{i+1} \end{cases}$$

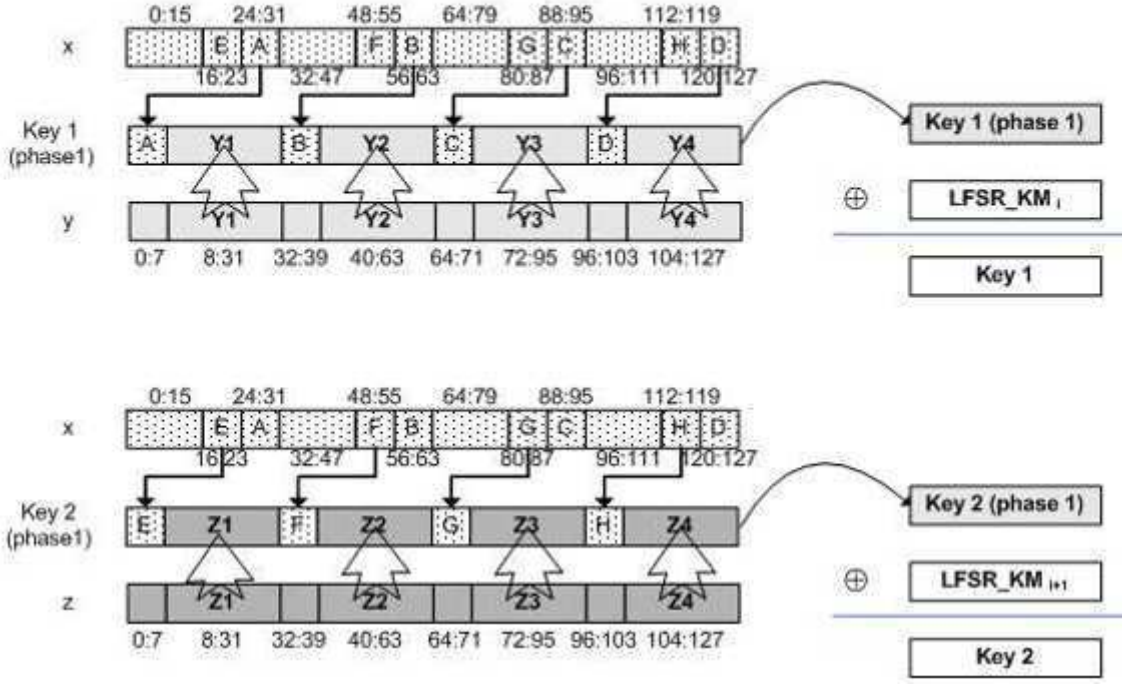

Figura 9: Funcionamento do Sub-Sistema de Mistura de Chaves [7]

4. IMPLEMENTAÇÃO DE CIFRAS CAÓTICAS

4.1. Critérios de seleção de Sistemas Caóticos

O mais importante na seleção dos sistemas caóticos para este trabalho, não é se estes são precisos, se têm precisão, mas sim a imprecisão caótica. Ou seja, dados os parâmetros de entrada corretos no sistema caótico, este fica completamente dessincronizado e desordenado e, assim, podemos garantir a criação de chaves realmente aleatórias e imprecisas.

São preferíveis sistemas de três dimensões para que seja mais fácil e consistente manter a imprecisão dos resultados. Outra boa característica dos sistemas é não terem equações muito longas, quando estas são curtas é mais fácil controlar a incerteza do sistema.

Os sistemas também precisam de continuar funcionais e caóticos depois de aplicado o método de Euler para os discretear, se estes não ficarem caóticos, não devolverem resultados inteiros ou aleatórios, não têm utilidade.

4.2. Sistemas Caóticos escolhidos

Dos sistemas caóticos existentes foram escolhidos alguns para serem implementados. Alguns deles foram rejeitados por não obedecerem as características especificadas, outros foram rejeitados após a implementados por não resultarem da forma pretendida e outros foram implementados com sucesso e mantiveram as condições necessárias. [32]

4.2.1. Não implementados

- **Sistemas/Mapas de duas dimensões** – Nenhum sistema de duas dimensões foi usado devido ao facto de ser mais complicado aplicar a arquitetura de caos modular e manter o caos no sistema com aplicação do método de Euler se este for um sistema dinâmico continuo porque quase todos são discretos. Por exemplo,

Bogdanov Map [41], Duffing Map [42], Gauss Map [43], tinkerbell Map [44], entre outros.

- Alguns sistemas não foram escolhidos devido ao facto de conterem equações de grande extensão. Por exemplo, Modified Lu Chen attractor [18], Modified Chua chaotic attractor [13, 18], PWL Duffing chaotic attractor [18], entre outros [18].
- Noutros sistemas, existia informação da sua existência, mas não havia qualquer informação da sua constituição, composição do sistema de equações. Por exemplo, Hadley cell [9], Lotka–Volterra equations [31], Nosé–Hoover thermostat [40], Rayleigh–Bénard convection [28] e Cellular neural network [36].

4.2.2. Implementados, mas rejeitados

- **Sistema Thomas Cyclical Symetric Atractor** – este sistema foi implementado com a parametrização b=0.2 mas, devido ao facto do mesmo funcionar em função do seno [14], este tornou-se impossível de implementar com a biblioteca GMP (*GNU Multiple Precision Arithmetic Library*). [15]

$$\begin{cases} \frac{dx}{dt} = \sin(y) - bx \\ \frac{dy}{dt} = \sin(z) - by \\ \frac{dz}{dt} = \sin(x) - bz \end{cases}$$

Figura 10: Thomas Cyclical Symetric Attractor [15]

4.2.3. Implementados

- **Sistema de Lorenz** – foi implementado com a parametrização de ρ=28, σ=10, e β=8/3. [10]

$$\begin{cases} \frac{\mathrm{d}x}{\mathrm{d}t} = \sigma(y-x), \\ \frac{\mathrm{d}y}{\mathrm{d}t} = x(\rho-z)-y, \\ \frac{\mathrm{d}z}{\mathrm{d}t} = xy-\beta z. \end{cases}$$

Figura 11: Sistema de Lorenz [10]

- **Sistema de Rossler attractor** – foi implementado com alguma dificuldade devido ao facto de ser implementado com parametrização de números decimais, com a parametrização a=0.1, b=0.1 e c=14. [16]

$$\begin{cases} \frac{dx}{dt} = -y - z \\ \frac{dy}{dt} = x + ay \\ \frac{dz}{dt} = b + z(x - c) \end{cases}$$

Figura 12: Sistema de Rossler attractor [16]

- **Sistema de Hindmarsh-Rose neuronal model** – foi implementado com o mesmo grau de dificuldade que o Sistema de Rossler attractor devido ao facto de ter parametrização decimal. Tem a seguinte parametrização s = 4, Xr = -8/5, a= 1, *b* = 3, *c* = 1, e *d* = 5 e *I* =3. [12]

$$\begin{cases} \frac{dx}{dt} = y + \phi(x) - z + I, \\ \frac{dy}{dt} = \psi(x) - y, \\ \frac{dz}{dt} = r[s(x - x_R) - z], \end{cases}$$

$$\phi(x) = -ax^3 + bx^2,$$
$$\psi(x) = c - dx^2.$$

Figura 13: Sistema Hindmarsh-Rose neuronal model [12]

- **Sistema de Rabinovich-Fabrikant Chaotic Attracctor** – este sistema tem parametrização toda decimal, mas foi escolhido exatamente por essa razão, para

se verificar se seria possível manter o caos no sistema mesmo com números arredondados, ou fracionados para inteiros, na parametrização. Tem a seguinte parametrização $\gamma = 0.1$ e $\alpha = 0.98$. [11]

$$\begin{cases} \dot{x} = y(z - 1 + x^2) + \gamma x \\ \dot{y} = x(3z + 1 - x^2) + \gamma y \\ \dot{z} = -2z(\alpha + xy), \end{cases}$$

Figura 14: Sistema de Rabinovich-Fabrikant [11]

4.3. Implementação

Toda a implementação dos sistemas caóticos foi realizada em linguagem C, em conjunto com a biblioteca de funções GMP, utilizando as funções para números "mpz". Estas são utilizadas devido ao facto de se estar a trabalhar com número extremamente grandes, de tamanho 2^{128}, e as variáveis inteiras da linguagem C não suportarem esses tamanhos extremos. [17]

A biblioteca GMP possui funções para todo o tipo de operações matemáticas e ainda outras funções para operações entre números "mpz" e números "int". Esta foi fundamental em todas as implementações, visto que, sem esta biblioteca nunca seria possível realizar as implementações. [17]

A implementação dos sistemas caóticos foi feita em aritmética modular 2^{128-1} com base na Arquitetura de Caos Modular, descrita no capítulo anterior, com a entrada dos parâmetros e a saída dos resultados por ficheiros ".txt". [37]

4.4. Recolha de Resultados

Para a recolha dos dados estatísticos, as implementações tiveram de ser modificadas para permitir criar chaves indefinidamente e guardá-las num ficheiro ".txt". Depois dessas modificações, decidiu-se de que tamanho seria o espaço de amostra de chaves. Chegou-se à conclusão que seria melhor ter uma amostra maior, de 1.000.000 de chaves em vez de 10.000 ou 100.000, para cada semente, existindo duas sementes de 16

caracteres, diferentes apenas num caractere, para se verificar a diversidade de resultados com o mínimo de diferença existente nas sementes.

Para que a recolha de chaves seja feita, garantidamente, no momento em que o sistema está caótico, foi inserido um salto das primeiras 1000 iterações de forma a que, quando este começa a recolher as chaves, já esteja com condições caóticas.

As chaves recolhidas são constituídas por 0's e 1's, cada chave possuindo 128 bits, sendo cada bit um 1 ou um 0. Foram recolhidas 1.000.000 de chaves para cada semente, das duas sementes de cada sistema caótico.

5. AVALIAÇÃO E COMPARAÇÃO DE RESULTADOS

5.1. Bateria de Testes

A Bateria de Testes utilizada nesta dissertação foi a NIST 800-22, *Statistical Test Suite for Random and Pseudo-Random Number Generators*. Esta é constituída por 15 testes diferentes especializados em diferentes métodos de teste de aleatoriedade. [19, 20, 21, 45] Os testes utilizados foram os seguintes:

1. *The Frequency (Monobit) Test,*
2. *Frequency Test Within a Block,*
3. *The runs Test,*
4. *Tests for the Longest-Run-of-Ones in a block,*
5. *The Binary Matrix Rank-Test,*
6. *The Discrete Fourier Transform (Spectral) Test,*
7. *The Non-Overlapping Template Matching Test,*
8. *The Overlapping Template Matching Test,*
9. *Maurer's "Universal Statistical" Test,*
10. *The Linear Complexity,*
11. *The Serial Test,*
12. *The Approximate Entropy Test,*
13. *The cumulative Sums (Cusums) Test,*
14. *The Random Excursion Test,*
15. *The Random Excursion Variant Test.*

5.2. Procedimentos para Teste

Após a realização de todos os testes, verificou-se que alguns não funcionavam da forma esperada para a amostra selecionada ou demonstravam, outros problemas.

5.2.1. Teste Realizados e Rejeitados

Este foram os testes que não foi possível executar, ou finalizar por razões diferentes, explicadas para cada um deles.

- 5 – **The Binary Matrix Rank-Test** foi realizado com várias entradas, 128 bits para 1 milhão sequências, 1 milhão de bits para 128 sequências, 1000 bits para 128000 sequências e 128000 bits para 1000 sequências, mas nenhum deu resultados favoráveis, sendo estes nulos. Isto aconteceu devido ao facto de este teste estar implementado com certos valores inalteráveis, resultantes da configuração do teste na implementação da bateria de testes.
- 7 – **The Non-Overlapping Template Matching Test** não foi possível finalizar, porque o Sistema em que foi testado tinha pouco espaço e este precisava de mais espaço para acabar e não houve outra oportunidade para testar noutra máquina.
- 8 – **The Overlapping Template Matching Test** não foi possível finalizar, porque o Sistema em que foi testado tinha pouco espaço e este precisava de mais espaço para acabar e não houve outra oportunidade para testar noutra máquina.
- 9 – **Maurer's "Universal Statistical" Test** não foi aplicado pois a configuração do mesmo não estava correta. O teste foi realizado segundo os dados que estavam referidos no documento da NIST (*National Institute of Standart and Tecnology*) e não dava resultados satisfatórios, tudo a 0 num milhão de valores, o que seria impossível.
- 14 – **The Random Excursion Test** só testava uma parte dos valores, por exemplo, o teste era realizado com 1 milhão de bits para 128 chaves e este não dava resposta nem para 10% das chaves.
- 15 – **The Random Excursion Variant Test** só testava uma parte dos valores, por exemplo, o teste era realizado com 1 milhão de bits para 128 chaves e este não dava resposta nem para 10% das chaves.

5.2.2. Testes Realizados e Aprovados

Estes foram os testes realizados, todos eles com base no documento oficial, NIST 800-22, da NIST. Apesar da bateria de teste utilizada não ser implementada pela NIST,

esta parece cumprir com os parâmetros descritos no documento, mas de uma forma um pouco rígida, estática, com alguns valores de entrada muito específicos.

- 1 – **The Frequency (Monobit) Test,** foi elaborado com os valores de entrada de chave de 128 bits para 1 milhão de chaves.
- 2 – **Frequency Test Within a Block** foi realizado com valores de entrada de chave de 128 bits para um milhão de chaves, o valor de M (tamanho do bloco) utilizado foi cde 16 bits mas foi também testado com 32 bits.
- 3 – **The runs Test** foi elaborado com os valores de entrada de 128 bits para 1 milhão de chaves.
- 4 – **Tests for the Longest-Run-of-Ones in a block** foi executado com os valores de entrada de chave de 128 bits para 1 milhão de chaves.
- 6 – **The Discrete Fourier Transform (Spectral) Test** foi realizado com os valores de entrada de chave de 128 bits para 1 milhão de chaves.
- 10 – **The Linear Complexity** foi executado com tamanho de chave de 128 bits para 1 milhão de chaves e um valor de M igual a 8 bits.
- 11 – **The Serial Test** foi elaborado com tamanho de chave de 128 bits para 1 milhão de chaves e um valor de M igual a 8 bits.
- 12 – **The Approximate Entropy Test** foi executado com tamanho de chave 128 bits para 1 milhão de chaves e um valor de M igual a 1.
- 13 – **The cumulative Sums (Cusums) Test** foi concebido com os valores de entrada de 128 bits por chave, com 1 milhão de chaves.

5.3. Resultados

Nesta secção serão apresentados os dados recolhidos durante a elaboração desta dissertação.

5.3.1. Condições iniciais

Vão ser apresentadas comparações dos 20 valores iniciais, com um salto dos primeiros 1000 valores, para cada conjunto de duas sementes de cada sistema caótico.

A tabela 1 e figura 15 correspondem à comparação de sementes do Sistema Caótico de Lorenz.

Tabela 1: Comparação de sementes do Sistema Caótico de Lorenz

	Seed 1 - Lore		Seed 2 - Lore
1	1,75E+38	1	2,38933E+38
2	1,61962E+38	2	2,21152E+38
3	2,74384E+38	3	2,94933E+37
4	6,34198E+37	4	2,96543E+38
5	1,73937E+38	5	1,93465E+38
6	1,49442E+38	6	1,23266E+37
7	1,83718E+38	7	2,17564E+36
8	2,00856E+38	8	2,00472E+38
9	2,79849E+37	9	1,31473E+38
10	2,45278E+38	10	1,52801E+38
11	1,33273E+38	11	3,65236E+37
12	2,59375E+38	12	2,25621E+38
13	3,26536E+37	13	8,95492E+38
14	1,7517E+38	14	1,9099E+38
15	8,22679E+37	15	2,33735E+38
16	2,55804E+38	16	1,85987E+38
17	1,41859E+38	17	1,19107E+37
18	3,03544E+38	18	2,85522E+37
19	2,94832E+37	19	3,27309E+38
20	6,32584E+37	20	5,47652E+37

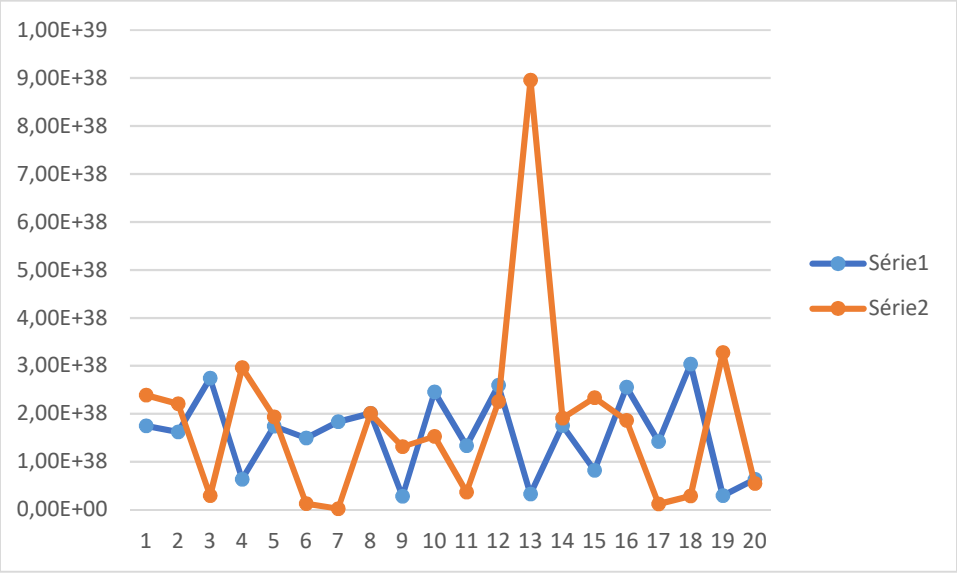

Figura 15: Gráfico de Sementes do Sistema Caótico de Lorenz

A tabela 2 e figura 16 correspondem à comparação de sementes do Sistema Caótico de Hindmarsh-Rose.

Tabela 2: Comparação de sementes do Sistema Caótico de Hindmarsh-Rose

	Seed 1 - Hind		Seed 2 - Hind
1	3,0052E+37	1	9,78354E+38
2	3,22078E+38	2	2,10039E+37
3	1,09864E+38	3	2,48434E+38
4	2,89541E+38	4	2,98417E+37
5	2,90321E+38	5	2,70133E+38
6	3,1796E+38	6	1,74279E+38
7	7,39094E+37	7	3,07127E+38
8	4,75885E+37	8	2,68541E+38
9	2,01911E+38	9	2,41274E+38
10	3,0765E+38	10	4,08831E+37
11	2,52853E+38	11	4,038E+37
12	3,30463E+38	12	1,2387E+38
13	1,13392E+38	13	5,06364E+37
14	3,03268E+38	14	2,4597E+38
15	1,44988E+38	15	3,04425E+38
16	2,73001E+37	16	1,26434E+37
17	6,08376E+37	17	1,63171E+38
18	1,84609E+38	18	1,88588E+38
19	9,82862E+38	19	3,00324E+38
20	3,23629E+38	20	6,24058E+35

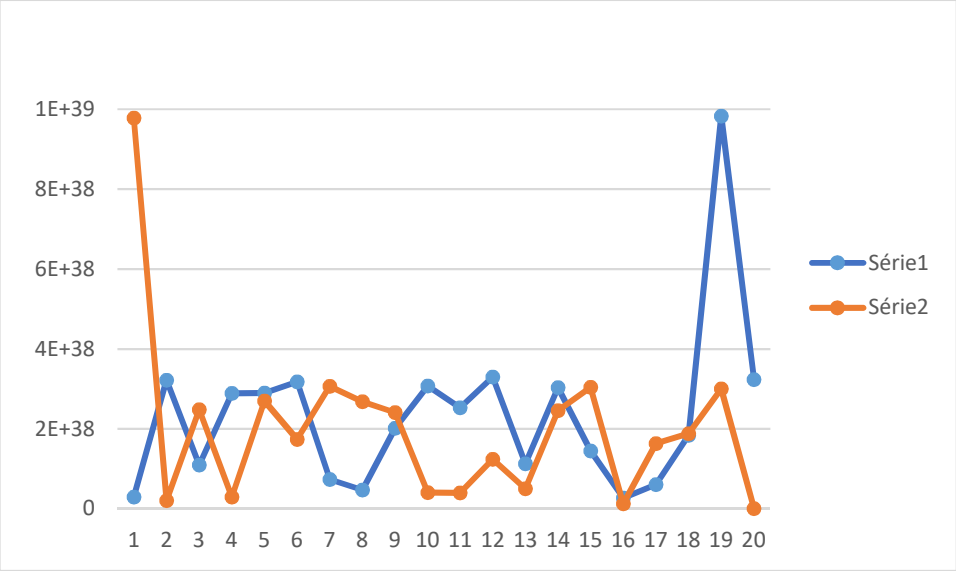

Figura 16: Gráfico de Sementes do Sistema Caótico de Hindmarsh-Rose

A tabela 3 e figura 17 correspondem à comparação de sementes do Sistema Caótico de Rossler.

Tabela 3: Comparação de sementes do Sistema Caótico de Rossler

	Seed 1 - Ross		Seed 2 - Ross
1	3,08874E+38	1	2,66733E+38
2	1,08806E+37	2	2,94203E+38
3	4,00002E+37	3	8,2634E+37
4	4,79139E+37	4	9,71446E+38
5	2,9379E+38	5	2,73876E+38
6	1,7348E+38	6	1,45578E+38
7	2,94764E+38	7	1,83115E+38
8	2,41093E+38	8	1,67988E+38
9	1,76374E+38	9	2,116E+38
10	6,49254E+37	10	2,20586E+37
11	1,34813E+38	11	1,9978E+38
12	1,95544E+37	12	1,91607E+38
13	5,78953E+37	13	3,07151E+38
14	2,01614E+38	14	1,888E+38
15	3,08167E+38	15	2,47926E+38
16	1,18194E+38	16	1,99729E+38
17	3,25125E+37	17	2,78646E+38
18	1,55825E+38	18	3,14781E+38
19	3,80358E+37	19	1,35458E+38
20	2,44987E+38	20	4,24738E+37

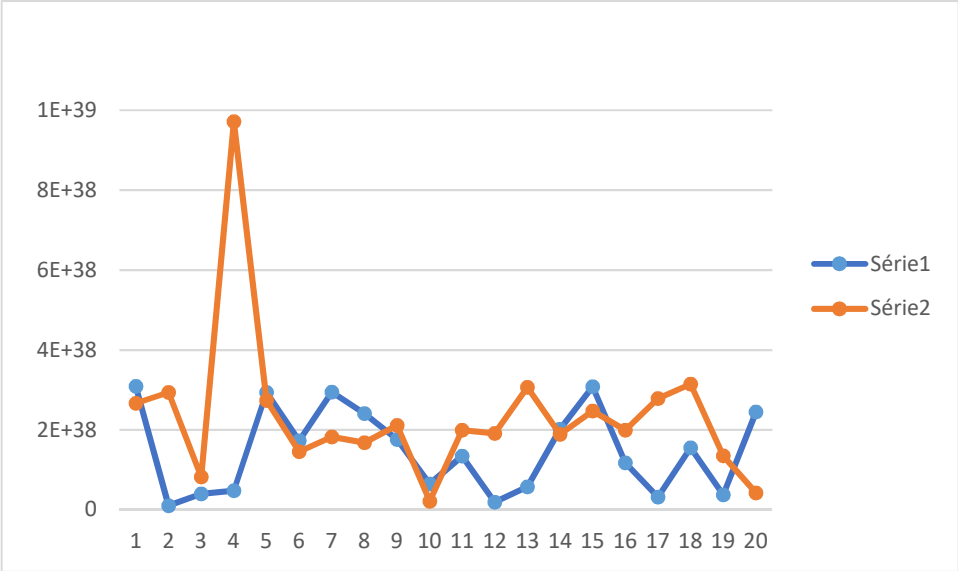

Figura 17: Gráfico de Sementes do Sistema Caótico de Rossler

Para finalizar, a tabela 4 e figura 18 correspondem à comparação de sementes do Sistema Caótico de Rabinovich-Fabrikant.

Tabela 4: Comparação sementes do Sistema Caótico de Rabinovich -Fabrikant

	Seed 1 - Rabi		Seed 2 - Rabi
1	1,34213E+38	1	1,02687E+38
2	2,45714E+37	2	1,74962E+38
3	2,49553E+38	3	2,98828E+38
4	2,26462E+38	4	1,52072E+38
5	1,47127E+38	5	2,96023E+38
6	1,68603E+38	6	3,0619E+38
7	4,11016E+37	7	1,65058E+38
8	3,14975E+38	8	2,66627E+38
9	2,39E+37	9	2,72882E+38
10	5,98022E+37	10	2,80662E+38
11	2,06688E+38	11	6,07921E+37
12	3,12698E+38	12	3,82641E+37
13	2,93259E+38	13	4,97043E+37
14	2,75565E+38	14	2,06292E+38
15	2,04244E+38	15	4,06722E+37
16	2,16435E+38	16	6,61945E+37
17	2,21867E+38	17	2,11214E+38
18	2,36545E+38	18	3,0632E+38
19	8,63712E+38	19	1,35912E+37
20	7,96108E+37	20	2,81281E+38

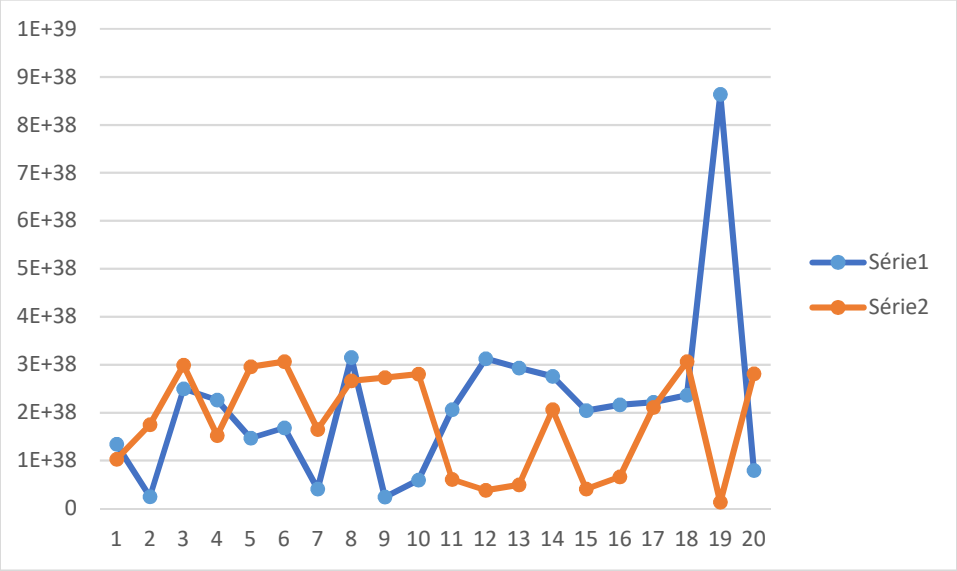

Figura 18: Gráfico de Sementes do Sistema Caótico de Rabinovich-Fabrikant

5.3.2. Balanço Binário

O balanço binário é a contabilização dos 0‘s e dos 1’s existentes na amostra recolhida de 1 milhão de chaves para cada semente. Um gerador de números pseudoaleatórios deve apresentar uma distribuição equilibrada entre zeros e uns.

A tabela 5 tem a conclusão do balanço binário dos dados recolhidos para realização dos testes.

Tabela 5: Balanço Binário dos 4 Sistemas Caóticos

Sistema Caótico	Seed	Balanço Binário
Lorenz	seed1	63973713 : 64026287
	seed2	63996144 : 64003856
Rossler	seed1	63932594 : 64067406
	seed2	64101448 : 63898552
Hindmarsh-Rose	seed1	64043052 : 63956948
	seed2	64015474 : 63984526
Rabinovich-Fabrikant	seed1	64011367 : 63988633
	seed2	63958791 : 64041209

5.3.3. Entropia Média

A Entropia Média é a análise da relação entre duas chaves consecutivas, ou seja, a verificação da existência de uma diferença média de 64 bits, dos 128 bits de cada chave, precisamente 50%, como é suposto. [46]

Nas figuras 19, 20, 21 e 22 verifica-se que a Entropia Média de cada sistema caótico está correta e dentro da conformidade média dos 50%.

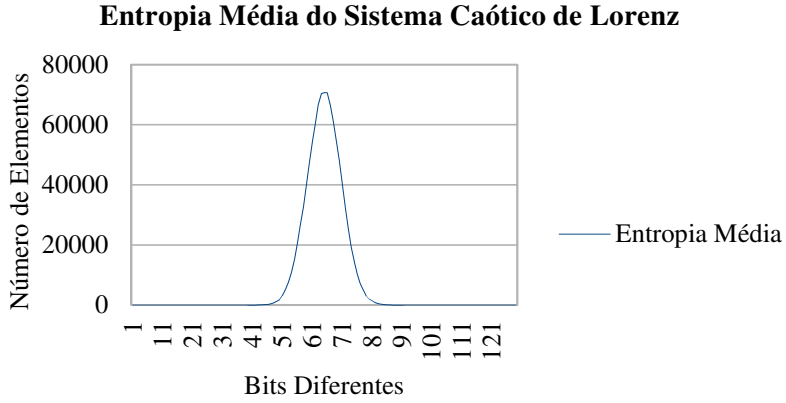

Figura 19: Entropia Média do Sistema Caótico de Lorenz

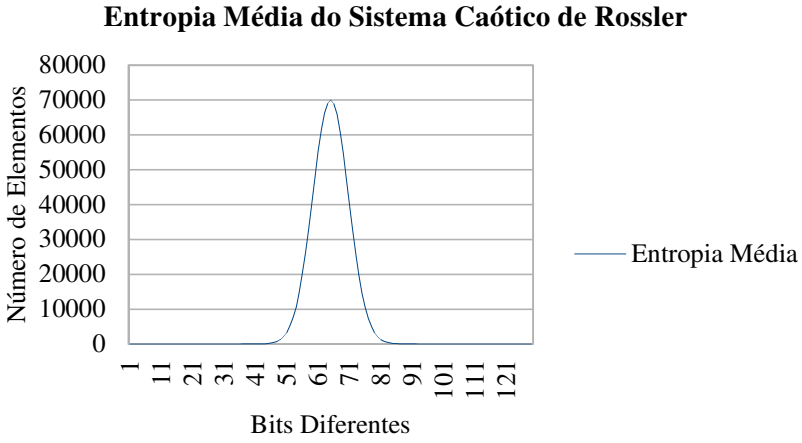

Figura 20: Entropia Média do Sistema Caótico de Rossler

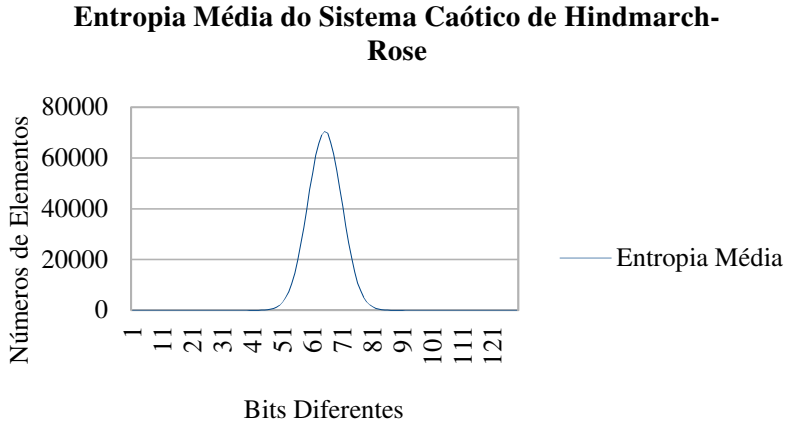

Figura 21: Entropia Média do Sistema Caótico de Hindmarch-Rose

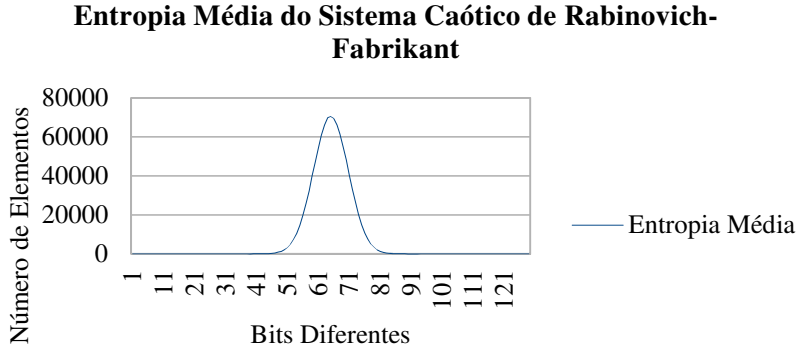

Figura 22: Entropia Média do Sistema Caótico de Rabinovich-Fabrikant

5.3.4. Resultados da Bateria de Testes

Nas tabelas 6, 7, 8, 9, 10, 11, 12 e 13 são apresentados os resultados individuais de cada semente para cada sistema caótico.

Tabela 6: Resultados Semente 1 Sistema Caótico de Lorenz

C1	C2	C3	C4	C5	C6	C7	C8	C9	C10	P-Value	Proporção estatistica	Testes
89922	90128	146212	96301	110004	122431	0	133123	140254	71625	0.077497	990290/1000000	Frequency
93814	95329	101667	104910	101189	110026	89037	106700	108758	88570	0.081211	992423/1000000	BlockFrequency
96938	102901	113863	102487	95893	104203	71620	110398	115122	86575	0.532041	990431/1000000	Runs
94662	103738	101799	106469	103816	120817	77558	108834	89017	93290	0.507459	991113/1000000	LongestRun
122412	115268	0	211689	0	245920	0	0	304711	0	0,500006	983744/1000000	FFT
112419	72044	87471	100165	80600	783191	89880	119862	151821	107347	0,500006	967727/1000000	LinearComplexity
101551	92159	97203	108739	101224	102119	89216	108659	109837	89293	0.301535	987465/1000000	Serial
101520	101750	96393	110982	99270	90066	103680	97506	100547	98286	0.709004	990235/1000000	Serial
102556	94483	92250	111688	91195	120005	77495	120695	85326	104307	0.150710	990479/1000000	ApproximateEntropy
80895	100488	79132	103582	129664	78710	77350	86391	155033	108755	0.125503	990880/1000000	CumulativeSums
81297	99918	78844	104366	130213	78170	76036	85994	155330	109832	0.067263	990804/1000000	CumulativeSums

Tabela 7: Resultados Semente 2 Sistema Caótico de Lorenz

C1	C2	C3	C4	C5	C6	C7	C8	C9	C10	P-Value	Proporção estatistica	Testes
92944	90088	145513	94893	110679	123596	0	133064	138449	70774	0.597746	988928/1000000	Frequency
93568	97307	104013	106550	103104	109850	87591	106076	106205	85736	0.999743	992631/1000000	BlockFrequency
101030	101846	112876	101471	96245	103949	71395	110090	114925	86173	0.569583	989915/1000000	Runs
93420	104171	101109	105543	102841	120316	78141	110549	89407	94503	0.309152	990717/1000000	LongestRun
121829	112790	0	213667	0	244373	0	0	307341	0	0,338114	984047/100000	FFT
111713	72321	86674	100715	80677	78648	89637	120968	151404	107243	311720	967608/100000	LinearComplexity
100367	89130	94859	107849	101149	102410	90599	110255	112860	90522	0.506360	988162/1000000	Serial
95269	99571	93400	109519	101140	91240	103549	99102	103149	104061	0.294269	990218/1000000	Serial
105578	95294	93022	109871	90570	119267	76994	120974	85511	102919	0.811546	988966/1000000	ApproximateEntropy
83128	100897	78534	103092	130768	78614	76576	86652	155302	106437	0.744756	989680/1000000	CumulativeSums
83010	101053	79486	103192	129391	77891	76690	87063	154899	107325	0.744756	989593/1000000	CumulativeSums

Tabela 8: Resultados Semente 1 Sistema Caótico de Rossler

C1	C2	C3	C4	C5	C6	C7	C8	C9	C10	P-Value	Proporção estatistica	Testes
92091	92248	146706	95483	109984	122630	0	131943	138594	70321	0.284789	989951/1000000	Frequency
94791	97799	102452	105000	102225	109446	88558	105846	107073	86810	0.383536	992545/1000000	BlockFrequency
102841	102055	112842	101704	95450	103476	71824	110443	114028	85337	0.807379	988754/1000000	Runs
95842	104681	102125	106129	103517	120367	77120	108789	88649	92781	0.192445	990579/1000000	LongestRun
122079	114469	0	212499	0	244968	0	0	305985	0	874706	984341/1000000	FFT
111517	72212	87184	100639	80867	79015	89757	199996	151192	107621	0,499508	968082/1000000	LinearComplexity
101650	91845	96247	108132	101312	102471	89209	108733	110517	89884	0.063447	987201/1000000	Serial
99539	101420	95894	111126	99755	90302	103121	97172	101276	100395	0.034800	989988/1000000	Serial
107347	95630	93267	110566	90807	119539	77468	119024	83996	102356	0.556439	989044/1000000	ApproximateEntropy
82724	102519	79820	104321	130365	78748	76768	86395	153953	104387	0.313109	990703/1000000	CumulativeSums
82704	102230	79916	104982	130426	78645	76996	86597	153130	104374	0.313109	990703/1000000	CumulativeSums

Tabela 9: Resultados Semente 2 Sistema Caótico de Rossler

C1	C2	C3	C4	C5	C6	C7	C8	C9	C10	P-Value	Proporção estatistica	Testes
92885	91702	146243	94911	109819	122816	0	132976	138433	70215	0.870238	989742/1000000	Frequency
96229	97319	103008	104881	101486	108306	87811	105902	107263	87795	0.249219	992090/1000000	BlockFrequency
102389	102195	112850	101562	96050	103666	71628	110183	114057	85420	0.860449	989007/1000000	Runs
96399	104820	101467	107091	102290	119598	77608	108583	89038	93106	0.334457	990697/1000000	LongestRun
121679	114941	0	212785	0	243942	0	0	306653	0	0,874706	984173/1000000	FFT
112561	72572	87971	101385	81112	78908	89512	118959	150314	106706	0,281024	968204/1000000	LinearComplexity
102392	91527	95837	107845	101177	102580	88756	109102	110130	90654	0.791800	987176/1000000	Serial
99036	101351	95383	110472	100524	90004	103493	97103	101364	101270	0.728397	990013/1000000	Serial
107727	95647	92647	110099	90842	119340	77004	119168	84518	103008	0.995839	989088/1000000	ApproximateEntropy
83858	101398	80344	104014	131060	78339	76658	85796	153528	105005	0.500000	990425/1000000	CumulativeSums
83859	101767	79724	103942	129932	78900	76211	86757	154123	104785	0.649879	990382/1000000	CumulativeSums

Tabela 10: Resultados Semente 1 Sistema Caótico de Hindmarsh-Rose

C1	C2	C3	C4	C5	C6	C7	C8	C9	C10	P-Value	Proporção estatistica	Testes
94264	92557	146275	95531	109632	121545	0	131763	138163	70270	0.284789	989662/1000000	Frequency
96504	98399	103818	105908	102642	108817	87202	105001	105796	85913	0.533289	992199/1000000	BlockFrequency
98427	100203	113043	102276	95937	104656	72447	111236	115204	86571	0.537665	989867/1000000	Runs
96353	104461	101470	106762	102782	119921	77638	109114	88985	92514	0.405744	990598/1000000	LongestRun
121277	114876	0	212397	0	246188	0	0	305262	0	874706	984630/1000000	FFT
112463	72718	87104	100507	80948	78443	89593	119486	151027	107711	0,333684	967776/1000000	LinearComplexity
101165	92752	96575	107966	101110	102613	89192	108548	109750	90329	0.063447	987688/1000000	Serial
98711	101449	96385	111149	100245	89947	103312	97279	101178	100345	0.026774	990166/1000000	Serial
105502	95792	92747	110246	90714	120025	77781	119600	84331	103262	0.469258	989476/1000000	ApproximateEntropy
84769	102862	80009	103986	130693	78867	76332	85928	152501	104053	0.222213	990549/1000000	CumulativeSums
84732	103009	79924	104690	130394	78549	76499	85728	152657	103818	0.372774	990502/1000000	CumulativeSums

Tabela 11: Resultados Semente 2 Sistema Caótico de Hindmarsh-Rose

C1	C2	C3	C4	C5	C6	C7	C8	C9	C10	P-Value	Proporção estatistica	Testes
90081	91485	145830	95329	110458	122919	0	133369	139497	71032	0.284789	990619/1000000	Frequency
92836	97120	102844	104685	102899	109930	89203	106300	107527	86656	0.751350	992930/1000000	BlockFrequency
99867	100834	113351	102060	96394	103984	71489	110849	115069	86103	0.125497	989711/1000000	Runs
94096	104749	101113	107217	103612	119428	77584	109380	89380	93441	0.791546	991091/1000000	LongestRun
1121324	115064	0	213020	0	244407	0	0	306185	0	0,331884	984360/1000000	FFT
112060	72505	87344	100876	80985	78618	89780	119327	151484	107021	372749	968272/1000000	LinearComplexity
98860	90986	96120	107621	101672	103014	90277	108996	111079	91375	0.812202	988027/1000000	Serial
98130	101284	95524	110788	100800	90026	103544	97613	100879	101412	0.874057	990295/1000000	Serial
102928	95449	91905	109888	91222	120248	78009	120481	85591	104279	0.176251	989940/1000000	ApproximateEntropy
81110	101250	79648	103693	131243	79175	76866	86850	154985	105180	0.222213	991252/1000000	CumulativeSums
80918	100541	79835	104510	130799	79613	77145	86824	154551	105264	0.372774	991299/1000000	CumulativeSums

Tabela 12: Resultados Semente 1 Sistema Caótico de Rabinovich-Fabrikant

C1	C2	C3	C4	C5	C6	C7	C8	C9	C10	P-Value	Proporção estatistica	Testes
91306	91280	146699	95876	109916	122182	0	132889	139007	70845	0.597746	990447/1000000	Frequency
95571	97312	102536	105446	101994	109122	88129	105877	107142	86871	0.623655	992391/1000000	BlockFrequency
101352	101322	112962	101875	95589	104419	72380	110510	113763	85828	0.569583	989154/1000000	Runs
95883	104773	101940	106584	103284	119201	77576	108468	89020	93271	0.649947	990809/1000000	LongestRun
121008	1154880	0	212028	0	245915	0	0	305561	0	0,523336	984455/1000000	FFT
112338	72765	87074	100823	81662	78675	89704	118677	150485	107797	0,644169	967962/1000000	LinearComplexity
100997	91866	96441	108003	101891	102471	88836	108609	110409	90477	0.352505	987527/1000000	Serial
99501	101653	96022	110412	100220	89798	102969	97375	101061	100989	0.170443	990080/1000000	Serial
105320	96271	93124	110896	90049	119954	77584	119733	84452	102617	0.811546	989786/1000000	ApproximateEntropy
82401	101710	79520	104653	131117	78735	76250	86710	153929	104975	0.941532	991129/1000000	CumulativeSums
82011	102510	80610	104085	130240	79601	76464	86061	153567	104851	0.500000	991271/1000000	CumulativeSums

Tabela 13: Resultados Semente 2 Sistema Caótico de Rabinovich-Fabrikant

C1	C2	C3	C4	C5	C6	C7	C8	C9	C10	P-Value	Proporção estatistica	Testes
91483	91645	145744	95072	109769	122885	0	133158	139293	70951	0.021493	989895/1000000	Frequency
93732	97057	102473	105058	102758	109911	88688	106662	107151	86510	0.249219	992830/1000000	BlockFrequency
100986	101421	113652	102801	95472	103704	71537	110883	113891	85653	0.075384	989430/1000000	Runs
94612	103522	101508	106650	103019	120381	77757	109353	89665	93533	0.053547	990909/1000000	LongestRun
121679	114941	0	212785	0	243942	0	0	306653	0	0.874706	984173/1000000	FFT
112561	72572	87971	101385	81112	78908	89512	118959	150314	106706	0.281024	968204/1000000	LinearComplexity
99361	91035	96623	107502	101569	102623	89247	110009	111128	90903	0.173246	987958/1000000	Serial
98373	100359	95019	111071	100060	90021	104337	97790	101684	101286	0.529456	990187/1000000	Serial
105485	95181	92732	110290	90508	120266	77096	119753	84975	103714	0.019489	989453/1000000	ApproximateEntropy
82290	101739	79616	103471	130782	78957	76821	86803	154770	104751	0.033700	990739/1000000	CumulativeSums
82460	101332	79237	104141	130015	79127	77093	86805	154925	104865	0.042986	990576/1000000	CumulativeSums

6. INTERPRETAÇÃO DE RESULTADOS E CONCLUSÕES FINAIS

Durante toda a implementação surgiram algumas dificuldades a nível de programação devido ao facto de estar a trabalhar com uma biblioteca, praticamente, nova para mim em que mais de metade das funções eram desconhecidas. Existiu a necessidade de explorar a biblioteca mais a fundo para a compreender.

Os Sistemas Caóticos implementados responderam ao esperado, estes, à entrada da mensagem, cifraram-na sem qualquer problema dando uma mensagem cifra da como resultado. Por sua vez, esta, continha caracteres estranho sem qualquer contexto em relação a inicial. Para garantir a funcionalidade do Sistema Caótico a mensagem cifra foi utilizada como mensagem inicial para verificar se o sistema decifrava. À entrada da mensagem cifra, esta depois de processada, deu resultado à mensagem inicial sem qualquer caractere de diferença. Funcionando assim para qualquer um dos Sistemas Caóticos.

Assim que ficou certificada a funcionalidade dos Sistemas Caóticos para cifragem e decifragem, começou outra batalha. A transformação para a recolha de chaves em binário para os testes estatísticos da NIST 800-22. A dificuldade foi conseguir transformar as chaves em linguagem máquina para números percetíveis com a biblioteca GMP. Depois de alcançado esse passo e feita a recolha de dados, foram realizados os testes estatísticos.

Os resultados dos testes estatísticos foram positivos tudo indica que as chaves recolhidas têm um comportamento aleatório. Dito isto, segundo a documentação da NIST o valor do *P value* define se a amostra é aleatória ou não. Para este caso, de 1 milhão de chaves, se o *P value* for maior ou igual a 0,000001, o teste é considerado positivo, o espaço de amostra é aleatório. Deste modo, todos os testes realizados às sementes dos Sistemas Caóticos deram positivo, e com base nesses resultados podemos dizer que caos e a desordem foram mantidos nos sistemas na recolha das chaves.

A comparação dos primeiros 20 resultados das sementes de cada Sistema Caóticos foi bastante positiva, devido ao facto, de que os resultados não se sobrepuseram, foram completamente diferentes para sementes com apenas 1 caractere de diferença. Foi mantido o estado caótico sem que os valores fossem iguais entre as sementes.

O balanço binário dos sistemas teve uma diferença, de 0's e 1's, menor que os 0,1%, à exceção do Sistema Caótico de Rossler que a diferença ficou um pouco mais acima, mas sem ultrapassar os 0,2%.

A Entropia Média em todos os Sistemas Caóticos manteve uma diferença média de 50%. Em que, na comparação dos resultados das chaves das sementes, de cada sistema, existe uma Entropia Média ideal de 50%, o que é excelente.

Com todo o estudo, conclui-se que a aplicação dos Sistemas Caóticos sobre Caos Modular à criptografia é viável, existe uma grande possibilidade de maior investimento neste tipo de cifras. Estas cifras podem ser aplicadas principalmente em dispositivos ou equipamentos que necessitem baixo custo computacional e energético. Sendo mantido o Caos nos Sistemas Caóticos durante toda a criação das chaves o resultado desta dissertação é bastante positivo para um futuro Sistema de Cifragem com base em Sistemas Caóticos.

6.1. Trabalhos Futuros

Ainda existe muitas possibilidades de desenvolvimento para Cifras Caóticas. Estas podem ser exploradas para reduzir ainda mais o custo computacional ou ser expostas a ataques de modo a explorar as suas fraquezas para que se possa melhorar e criar algo melhor. Tudo depende das necessidades futurar se criar algo mais robusto e com maior custo computacional e infalível ou algo de baixo custo computacional mas bastante fiável.

7. BIBLIOGRAFIA

[1] "Segurança: a evolução da tecnologia na proteção da sociedade," 2009. [Online]. Available: https://www.tecmundo.com.br/camera-digital/3192-seguranca-a-evolucao-da-tecnologia-na-protecao-da-sociedade.htm. [Acedido em 2017].

[2] "A lenta evolução da Internet das Coisas.," [Online]. Available: http://blogbrasil.comstor.com/a-lenta-evolucao-da-internet-das-coisas. [Acedido em 2017].

[3] Ticiano, "Segurança da Informação como Gestão de Riscos de Negócio," 2017. [Online]. Available: https://ticianobenetti.wordpress.com/. [Acedido em 2017].

[4] B. B. Worldwide, "Gráfico do Histórico de Preço do Bitcoin," [Online]. Available: https://www.buybitcoinworldwide.com/pt-br/preco/. [Acedido em 2017].

[5] J. L. Lulu Yilun Chen, "Uma decisão do banco central chinês derrubou o valor de moedas digitais como o bitcoin," 2017. [Online]. Available: http://www.gazetadopovo.com.br/economia/nova-economia/uma-decisao-do-banco-central-chines-derrubou-o-valor-de-moedas-digitais-como-o-bitcoin-emqtpul8yjvqzy4clup9nuzce. [Acedido em 2017].

[6] Wikipédia, "Chaos theory," [Online]. Available: https://en.wikipedia.org/wiki/Chaos_theory. [Acedido em 2016].

[7] R. M. S. Silva, "Enhanced Chaotic Stream Cipher for WSNs," 2009. [Online]. Available: http://ieeexplore.ieee.org/document/5438094/. [Acedido em 2016].

[8] C. A. Romagnolo, "O que é Criptografia?," 2017. [Online]. Available: https://www.oficinadanet.com.br/artigo/443/o_que_e_criptografia. [Acedido em 2017].

[9] Wikipédia, "Hadley cell," [Online]. Available: https://en.wikipedia.org/wiki/Hadley_cell. [Acedido em 2016/2017].

[10] Wikipédia, "Lorenz system," [Online]. Available: https://en.wikipedia.org/wiki/Lorenz_system. [Acedido em 2016/2017].

[11] Wikipédia, "Rabinovich–Fabrikant equations," [Online]. Available: https://en.wikipedia.org/wiki/Rabinovich%E2%80%93Fabrikant_equations. [Acedido em 2016/2017].

[12] Wikipédia, "Hindmarsh–Rose model," [Online]. Available: https://en.wikipedia.org/wiki/Hindmarsh%E2%80%93Rose_model. [Acedido em 2016/2017].

[13] Wikipédia, "Chua's circuit," [Online]. Available: https://en.wikipedia.org/wiki/Chua%27s_circuit. [Acedido em 2016/2017].

[14] Wikipédia, "Seno," [Online]. Available: https://pt.wikipedia.org/wiki/Seno. [Acedido em 2016/2017].

[15] Wikipédia, "Thomas' cyclically symmetric attractor," [Online]. Available: https://en.wikipedia.org/wiki/Thomas%27_cyclically_symmetric_attractor. [Acedido em 2016].

[16] Wikipédia, "Rössler attractor," [Online]. Available: https://en.wikipedia.org/wiki/R%C3%B6ssler_attractor. [Acedido em 2016/2017].

[17] Gnu, "GNU MP 6.1.2," [Online]. Available: https://gmplib.org/manual/index.html#Top. [Acedido em 2016/2017].

[18] Wikipédia, "Multiscroll attractor," [Online]. Available: https://en.wikipedia.org/wiki/Multiscroll_attractor. [Acedido em 2016].

[19] NIST, "A Statistical Test Suite for Random and Pseudorandom Number Generators for Cryptographic Applications," [Online]. Available: https://www.nist.gov/publications/statistical-test-suite-random-and-pseudorandom-number-generators-cryptographic?pub_id=906762. [Acedido em 2016].

[20] NIST, "A Statistical Test Suite for Random and Pseudorandom Number Generators for Cryptographic Applications," [Online]. Available: https://csrc.nist.gov/publications/detail/sp/800-22/rev-1a/final. [Acedido em 2016].

[21] NIST, "National Institute of Security and Technology," [Online]. Available: https://www.nist.gov/. [Acedido em 2016].

[22] Wikipédia, "Teoria do caos," [Online]. Available: https://pt.wikipedia.org/wiki/Teoria_do_caos. [Acedido em 2016/2017].

[23] Wikipédia, "Teoria da informação," [Online]. Available: https://pt.wikipedia.org/wiki/Teoria_da_informa%C3%A7%C3%A3o. [Acedido em 2017].

[24] Wikipédia, "Sistemas complexos," [Online]. Available: https://pt.wikipedia.org/wiki/Sistemas_complexos. [Acedido em 2016/2017].

[25] Wikipédia, "Sistema dinâmico não linear," [Online]. Available: https://pt.wikipedia.org/wiki/Sistema_din%C3%A2mico_n%C3%A3o_linear. [Acedido em 2016/2017].

[26] Wikipédia, "Sistema dinâmico discreto," [Online]. Available: https://pt.wikipedia.org/wiki/Sistema_din%C3%A2mico_discreto. [Acedido em 2016/2017].

[27] Wikipédia, "Sistema dinâmico contínuo," [Online]. Available: https://pt.wikipedia.org/wiki/Sistema_din%C3%A2mico_cont%C3%ADnuo.

[28] Wikipédia, "Rayleigh–Bénard convection," [Online]. Available: https://en.wikipedia.org/wiki/Rayleigh%E2%80%93B%C3%A9nard_convection. [Acedido em 2016/2017].

[29] Wikipédia, "Modo de operação (criptografia)," [Online]. Available: https://pt.wikipedia.org/wiki/Modo_de_opera%C3%A7%C3%A3o_(criptografia)#Modo_CTR_.28Counter.29:. [Acedido em 2017].

[30] Wikipédia, "Método de Euler," [Online]. Available: https://pt.wikipedia.org/wiki/M%C3%A9todo_de_Euler. [Acedido em 2016/2017].

[31] Wikipédia, "Lotka–Volterra equations," [Online]. Available: https://en.wikipedia.org/wiki/Lotka%E2%80%93Volterra_equations. [Acedido em 2016/2017].

[32] Wikipédia, "List of chaotic maps," [Online]. Available: https://en.wikipedia.org/wiki/List_of_chaotic_maps. [Acedido em 2016/2017].

[33] Wikipédia, "Linear-feedback shift register," [Online]. Available: https://en.wikipedia.org/wiki/Linear-feedback_shift_register. [Acedido em 2016/2017].

[34] Wikipédia, "Criptografia de chave pública," [Online]. Available: https://pt.wikipedia.org/wiki/Criptografia_de_chave_p%C3%BAblica.

[35] Wikipédia, "Criptografia," [Online]. Available: https://pt.wikipedia.org/wiki/Criptografia. [Acedido em 2016/2017].

[36] Wikipédia, "Cellular neural network," [Online]. Available: https://en.wikipedia.org/wiki/Cellular_neural_network. [Acedido em 2016/2017].

[37] Wikipédia, "Aritmética modular," [Online]. Available: https://pt.wikipedia.org/wiki/Aritm%C3%A9tica_modular. [Acedido em 2016/2017].

[38] Wikipédia, "Algoritmo de chave simétrica," [Online]. Available: https://pt.wikipedia.org/wiki/Algoritmo_de_chave_sim%C3%A9trica. [Acedido em 2016/2017].

[39] Wikipédia, "Fractal," [Online]. Available: https://en.wikipedia.org/wiki/Fractal. [Acedido em 2017].

[40] Wikipédia, "Nosé–Hoover thermostat," [Online]. Available: https://en.wikipedia.org/wiki/Nosé–Hoover_thermostat. [Acedido em 2016].

[41] Wikipédia, "Bogdanov map," [Online]. Available: https://en.wikipedia.org/wiki/Bogdanov_map. [Acedido em 2016].

[42] Wikipédia, "Duffing map," [Online]. Available: https://en.wikipedia.org/wiki/Duffing_map. [Acedido em 2016].

[43] Wikipédia, "Gauss iterated map," [Online]. Available: https://en.wikipedia.org/wiki/Gauss_iterated_map. [Acedido em 2016].

[44] Wikipédia, "Tinkerbell map," [Online]. Available: https://en.wikipedia.org/wiki/Tinkerbell_map. [Acedido em 2016].

[45] Wikipédia, "Pseudorandomness," [Online]. Available: https://en.wikipedia.org/wiki/Pseudorandomness. [Acedido em 2016].

[46] Wikipédia, "Entropia da informação," [Online]. Available: https://pt.wikipedia.org/wiki/Entropia_da_informa%C3%A7%C3%A3o. [Acedido em 2017].

[47] Wikipédia, "Exclusive or," [Online]. Available: https://en.wikipedia.org/wiki/Exclusive_or. [Acedido em 2016].

[48] Wikipédia, “Advanced Encryption Standard,” [Online]. Available: https://en.wikipedia.org/wiki/Advanced_Encryption_Standard. [Acedido em 2016].

8. ANEXOS

- Código versões iniciais sem GMP
- Código versões intermédias
- Código com GMP para cifrar e decifrar
- Código para recolha de chaves em binário
- Código para recolha de Chaves em decimal
- Chaves recolhidas com código incorreto
- Chaves recolhidas em binário e decimal
- Documento da NIST 800-22
- Resultados e Tabelas e Gráficos dos Resultados
- Código para funções de Entropia média, balanço binário

Todos os anexos acima referidos encontram-se em formato digital.